

43
SDF-0083

P-1857

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

2000年 6月22日

出 願 番 号

Application Number:

特願2000-188197

出 願 人

Applicant (s):

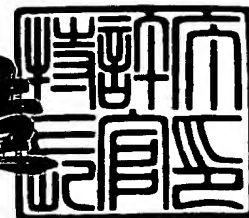
新光電気工業株式会社



2001年 3月 9日

特許庁長官
Commissioner,
Patent Office

及 川 耕 造



出証番号 出証特2001-3017773

【書類名】 特許願

【整理番号】 P0056180

【提出日】 平成12年 6月22日

【あて先】 特許庁長官 殿

【国際特許分類】 G03F 1/08

【発明の名称】 マイナスサイジング処理方法とコンピュータに実行させるためのプログラムを格納したコンピュータが読み取り可能な記憶媒体

【請求項の数】 4

【発明者】

【住所又は居所】 長野県長野市大字栗田字舎利田 7 1 1 番地 新光電気工業株式会社内

【氏名】 吉澤 恵資

【特許出願人】

【識別番号】 000190688

【氏名又は名称】 新光電気工業株式会社

【代理人】

【識別番号】 100077621

【弁理士】

【氏名又は名称】 綿貫 隆夫

【選任した代理人】

【識別番号】 100092819

【弁理士】

【氏名又は名称】 堀米 和春

【手数料の表示】

【予納台帳番号】 006725

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 マイナスサイジング処理方法とコンピュータに実行させるためのプログラムを格納したコンピュータが読み取り可能な記憶媒体

【特許請求の範囲】

【請求項 1】 元図形と該元図形に図形処理を施して生成された新図形、若しくは元図形に異なる処理システムを適用して生成された 2 つの新図形同士を重ね合わせた際に生ずる差分図形の大小に基づいて前記図形処理の検証を行うために前記差分図形に対して施されるマイナスサイジング処理方法において、

前記差分図形の各頂点の組みを第 1 頂点列として記憶する第 1 頂点列記憶ステップと、

前記差分図形の各辺上から内方にサイジング量だけ仮想直線を平行移動させ、該仮想直線で囲まれて成るオフセット図形を生成すると共に、該オフセット図形の各頂点の組を第 2 頂点列として記憶するオフセット図形生成ステップと、

前記差分図形の各頂点と該各頂点に対応した前記オフセット図形の各頂点とを結ぶオフセット推移線分を生成するオフセット推移線分生成ステップと、

前記差分図形の隣接する各頂点から伸びる 2 本の前記オフセット推移線分同士の第 1 交点の有無を検出する裏返し検出ステップと、

該第 1 交点で交差する 2 本の前記オフセット推移線分が共に接続する前記差分図形の辺と前記第 1 交点との間の距離に基づいて第 1 交点を、該距離の短い順にソートするソートステップと、

ソートされた順に前記第 1 交点を一つ特定し、特定した第 1 交点で交差する 2 本の前記オフセット推移線分が共に接続する前記オフセット図形の裏返し辺の各端部に接続されたオフセット図形の 2 本の隣接辺同士の第 2 交点を算出し、特定した第 1 交点で交差する 2 本のオフセット推移線分が共に接続する前記差分図形の頂点および前記オフセット図形の頂点を前記第 1 頂点列および前記第 2 頂点列から削除し、特定した第 1 交点を第 1 頂点列に差分図形の新たな頂点として記憶すると共に、前記第 2 交点を第 2 頂点列に、前記差分図形の新たな頂点に対応するオフセット図形の新たな頂点として記憶する頂点更新ステップとを具備することを特徴とするマイナスサイジング処理方法。

【請求項 2】 前記頂点更新ステップにおいて、前記隣接辺同士が平行で前記第 2 交点が存在しない場合には、

前記第 1 交点を通過し、かつ前記隣接辺と平行な第 2 仮想直線を生成すると共に、2 本の隣接辺の非裏返り辺側の各端部に位置するオフセット図形の各頂点から伸びる 2 本のオフセット推移線分と前記第 2 仮想直線との第 3 交点を求め、2 本の隣接辺の内、第 1 交点により近い第 3 交点を含むオフセット推移線分と接続された隣接辺を前記裏返り辺と合わせ、全体を新たな裏返り辺とすることを特徴とする請求項 1 記載のマイナスサイジング処理方法。

【請求項 3】 元図形と該元図形に図形処理を施して生成された新図形、若しくは元図形に異なる処理システムを適用して生成された 2 つの新図形同士を重ね合わせた際に生ずる差分図形の大小に基づいて前記図形処理の検証を行うために前記差分図形に対して施されるマイナスサイジング処理をコンピュータに実行させるためのプログラムを格納したコンピュータが読み取り可能な記憶媒体において、

前記差分図形の各頂点の組みを第 1 頂点列として記憶する第 1 頂点列記憶ステップと、

前記差分図形の各辺上から内方にサイジング量だけ仮想直線を平行移動させ、該仮想直線で囲まれて成るオフセット図形を生成すると共に、該オフセット図形の各頂点の組を第 2 頂点列として記憶するオフセット図形生成ステップと、

前記差分図形の各頂点と該各頂点に対応した前記オフセット図形の各頂点とを結ぶオフセット推移線分を生成するオフセット推移線分生成ステップと、

前記差分図形の隣接する各頂点から伸びる 2 本の前記オフセット推移線分同士の第 1 交点の有無を検出する裏返り検出ステップと、

該第 1 交点で交差する 2 本の前記オフセット推移線分が共に接続する前記差分図形の辺と前記第 1 交点との間の距離に基づいて第 1 交点を、該距離の短い順にソートするソートステップと、

ソートされた順に前記第 1 交点を一つ特定し、特定した第 1 交点で交差する 2 本の前記オフセット推移線分が共に接続する前記オフセット図形の裏返り辺の各端部に接続されたオフセット図形の 2 本の隣接辺同士の第 2 交点を算出し、特定

した第 1 交点で交差する 2 本のオフセット推移線分が共に接続する前記差分図形の頂点および前記オフセット図形の頂点を前記第 1 頂点列および前記第 2 頂点列から削除し、特定した第 1 交点を第 1 頂点列に差分図形の新たな頂点として記憶すると共に、前記第 2 交点を第 2 頂点列に、前記差分図形の新たな頂点に対応するオフセット図形の新たな頂点として記憶する頂点更新ステップとを含むことを特徴とするコンピュータに実行させるためのプログラムを格納したコンピュータが読み取り可能な記憶媒体。

【請求項 4】 前記頂点更新ステップにおいて、前記隣接辺同士が平行で前記第 2 交点が存在しない場合には、

前記第 1 交点を通過し、かつ前記隣接辺と平行な第 2 仮想直線を生成すると共に、2 本の隣接辺の非裏返り辺側の各端部に位置するオフセット図形の各頂点から伸びる 2 本のオフセット推移線分と前記第 2 仮想直線との第 3 交点を求め、2 本の隣接辺の内、第 1 交点により近い第 3 交点を含むオフセット推移線分と接続された隣接辺を前記裏返り辺と合わせ、全体を新たな裏返り辺とすることを特徴とする請求項 3 記載のコンピュータに実行させるためのプログラムを格納したコンピュータが読み取り可能な記憶媒体。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は、元図形とこの元図形に図形処理を施して生成された新図形、若しくは元図形に異なる処理システムを適用して生成された 2 つの新図形同士を重ね合わせた際に生ずる差分図形の大小に基づいて図形処理の検証を行うために差分図形に対して施されるマイナスサイジング処理方法に関する。また、この処理をコンピュータに実行させるためのプログラムを格納したコンピュータが読み取り可能な記憶媒体に関する。

【 0 0 0 2 】

【従来の技術】

L S I のマスクパターン処理等の図形処理では、図形処理後の図形データの異常を確認するために従来から次の 2 つの検証処理が行われている。

まず、第 1 の検証処理方法は、特開平8-160598号公報に開示されているように、元図形（設計データ）とこの元図形に対して図形処理（一例として光近接交差補正）を施して生成された新図形とを重ね合わせて、言い換えれば排他的論理和をとって、差分図形を得る。そしてこの差分図形の大小によって、大きければ図形処理に問題があり、図形処理した新図形に誤りがあると判断する。また、差分図形が規定値よりも小さければ図形処理には問題が無く、図形処理した新図形にも誤りが無い、つまり正常であると判断する。図 7 参照。

【 0 0 0 3 】

第 2 の検証処理方法は、元図形に対して処理目的は同じであるが処理内容や手順の異なる別々の処理システムを適用し、各処理システムによって生成された 2 つの新図形同士に対して第 1 の検証処理方法と同様に差分図形を求め、この差分図形の大小によって、大きければいずれか一方の図形の処理システムに問題があり、どちらか一方の新図形に誤りがあると判断する。また、差分図形が規定値よりも小さければ、2 つの新図形は正常であると判断する。図 8 参照。

【 0 0 0 4 】

そして、差分図形が大きい小さいかを判断する手段としては、一般的にマイナスサイジング処理を差分図形に対して適用する手法が使用される。つまり、適当なサイジング量のマイナスサイジング処理を差分図形に対して施した結果、差分図形が消滅すれば、差分図形が小さく、一方、差分図形が消滅しなければ、差分図形が大きいと判断する。

ここでマイナスサイジング処理とは、差分図形等の元図形を構成する各辺上から仮想直線を所定のサイジング量（移動距離）だけ元図形の内方に平行移動（オフセット）させ、オフセット後の仮想直線同士の交点を結んで仮のオフセット図形を生成するオフセット処理工程と、この仮のオフセット図形に対して OR 処理を行い、正式なオフセット図形を生成する OR 処理工程とから成り、概念的には元図形をサイジング量分だけ細らせる処理である。よって、サイジング量（オフセット量とも言う）を元図形の寸法に比べて大きくしてマイナスサイジング処理を元図形に施すと、元図形は原則として消滅する。

【 0 0 0 5 】

しかしながら、図 9 や図 1 0 に示す形状の元図形 A (図 9 の場合：頂点 p_1, p_2, p_3, p_4) の場合、サイジング量が元図形 A に比べて大きく、本来であれば処理後に元図形 A が消滅しなければならないのに、オフセット処理を行った後の仮のオフセット図形 B (図 9 の場合：頂点 p_1', p_2', p_3', p_4') に OR 処理を施しても、裏返り辺の存在で消滅しないという現象が生ずる。ここで裏返り辺とは、元図形 A の一つの頂点とこれに対応する仮のオフセット図形 B の一つの頂点を結ぶ線分をオフセット線分とした場合、隣接するオフセット線分が交差する場合、このオフセット線分に接続される仮のオフセット図形の一辺を言う。

このため、本来であればマイナスサイジング処理の結果、元図形 A である差分図形が消滅して検証処理の結果が良好であると判断できるにもかかわらず、元図形 A が消滅せず、検証処理の結果が良好と判断できない場合が生ずる、という課題が生ずる。

【 0 0 0 6 】

そこで、従来から裏返り辺に起因する前記課題を解決できる種々の処理方法が考えられており、例えば図 1 1 に示す第 1 のサイジング処理方法では、元図形 A の各頂点 (p_1, p_2, p_3, p_4) に対してオフセット点を 2 点生成して処理を行うようにしている。例えば、頂点 p_1 が凸状である場合には、そのオフセット点は、辺 (p_1, p_2) をサイジング量だけ平行移動した際の頂点 p_1 の移動後の点と、辺 (p_4, p_1) をサイジング量だけ平行移動した際の頂点 p_1 の移動後の点である。また、頂点 p_1 が凹状である場合には、そのオフセット点は、辺 (p_1, p_2) 上からサイジング量だけ平行移動した仮想直線と、また辺 (p_4, p_1) 上からサイジング量だけ平行移動した他の仮想直線同士の交点となる。

また、第 2 のサイジング処理方法では、元図形 A (頂点： $p_1, p_2, p_3, p_4, p_5, p_6$) の各辺を一度にオフセットしてしまうと、図 1 2 のように仮のオフセット図形 B (頂点： $p_1', p_2', p_3', p_4', p_5', p_6'$) を生成した際に裏返り辺 (p_2', p_3')、(p_5', p_6') が生じてしまうため、図 1 3 に示すように元図形 A の各辺に対して小刻みにオフセットを繰り返して最終のオフセット図形を得る手法を採用している。

【 0 0 0 7 】

【発明が解決しようとする課題】

しかしながら、上記の改良されたマイナスサイジング処理には次のような課題がある。

まず、第1のサイジング処理方法では、元図形の各頂点に対して2つの頂点を新たに生成して処理を行うため、OR処理に長い時間を要するという課題がある。

具体的に、オフセット処理とOR処理にそれぞれ要する計算オーダを算出してみると、

オフセット処理： $O(N)$

OR処理： $O((aN) \log(aN))$

このとき $a \geq 2$

(形状の半分以上は凸頂点であり、オフセット後の頂点数は、凸頂点3倍、凹頂点1倍となる。マイナスサイジングの場合)

ここで、 \log は2を底とする。また、 N は元図形の頂点数、 a はOR処理への入力頂点数の増加率（OR処理への入力頂点数は aN となる）であるとなる。

【0008】

次に、第2のサイジング処理方法では、段階的にオフセット形状が求められるために、最終のオフセット点（オフセット図形の頂点）に含まれる計算誤差が増大してしまうという課題がある。

【0009】

そこで、本発明は上記課題を解決すべくなされたものであり、その目的とするところは、裏返り辺が発生する順に、裏返り辺の近傍のみ裏返り辺の除去処理を行うため、計算量が少なく高速であり、かつ正確なサイジングが行えるマイナスサイジング処理方法を提供することにある。

【0010】

【課題を解決するための手段】

すなわち、本発明に係る請求項1記載のマイナスサイジング処理方法は、元図形と該元図形に図形処理を施して生成された新図形、若しくは元図形に異なる処

理システムを適用して生成された 2 つの新図形同士を重ね合わせた際に生ずる差分図形の大小に基づいて前記図形処理の検証を行うために前記差分図形に対して施されるマイナスサイジング処理方法において、前記差分図形の各頂点の組みを第 1 頂点列として記憶する第 1 頂点列記憶ステップと、前記差分図形の各辺上から内方にサイジング量だけ仮想直線を平行移動させ、該仮想直線で囲まれて成るオフセット図形を生成すると共に、該オフセット図形の各頂点の組を第 2 頂点列として記憶するオフセット図形生成ステップと、前記差分図形の各頂点と該各頂点に対応した前記オフセット図形の各頂点とを結ぶオフセット推移線分を生成するオフセット推移線分生成ステップと、前記差分図形の隣接する各頂点から伸びる 2 本の前記オフセット推移線分同士の第 1 交点の有無を検出する交点検出ステップと、該第 1 交点で交差する 2 本の前記オフセット推移線分が共に接続する前記差分図形の辺と前記第 1 交点との間の距離に基づいて第 1 交点を、該距離の短い順にソートするソートステップと、ソートされた順に前記第 1 交点を一つ特定し、特定した第 1 交点で交差する 2 本の前記オフセット推移線分が共に接続する前記オフセット図形の裏返り辺の各端部に接続されたオフセット図形の 2 本の隣接辺同士の第 2 交点を算出し、特定した第 1 交点で交差する 2 本のオフセット推移線分が共に接続する前記差分図形の頂点および前記オフセット図形の頂点を前記第 1 頂点列および前記第 2 頂点列から削除し、特定した第 1 交点を第 1 頂点列に差分図形の新たな頂点として記憶すると共に、前記第 2 交点を第 2 頂点列に、前記差分図形の新たな頂点に対応するオフセット図形の新たな頂点として記憶する頂点更新ステップとを具備することを特徴とする。

【 0 0 1 1 】

また、前記頂点更新ステップにおいて、前記隣接辺同士が平行で前記第 2 交点が存在しない場合には、前記頂点更新ステップにおいて、前記隣接辺同士が平行で前記第 2 交点が存在しない場合には、前記第 1 交点を通過し、かつ前記隣接辺と平行な第 2 仮想直線を生成すると共に、2 本の隣接辺の非裏返り辺側の各端部に位置するオフセット図形の各頂点から伸びる 2 本のオフセット推移線分と前記第 2 仮想直線との第 3 交点を求め、2 本の隣接辺の内、第 1 交点により近い第 3 交点を含むオフセット推移線分と接続された隣接辺を前記裏返り辺と合わせ、全

体を新たな裏返り辺とすることを特徴とする。

【 0 0 1 2 】

また、請求項 3 記載のコンピュータに実行させるためのプログラムを格納したコンピュータが読み取り可能な記憶媒体によれば、元図形と該元図形に図形処理を施して生成された新図形、若しくは元図形に異なる処理システムを適用して生成された 2 つの新図形同士を重ね合わせた際に生ずる差分図形の大小に基づいて前記図形処理の検証を行うために前記差分図形に対して施されるマイナスサイジング処理をコンピュータに実行させるためのプログラムを格納したコンピュータが読み取り可能な記憶媒体において、前記差分図形の各頂点の組みを第 1 頂点列として記憶する第 1 頂点列記憶ステップと、前記差分図形の各辺上から内方にサイジング量だけ仮想直線を平行移動させ、該仮想直線で囲まれて成るオフセット図形を生成すると共に、該オフセット図形の各頂点の組を第 2 頂点列として記憶するオフセット図形生成ステップと、前記差分図形の各頂点と該各頂点に対応した前記オフセット図形の各頂点とを結ぶオフセット推移線分を生成するオフセット推移線分生成ステップと、前記差分図形の隣接する各頂点から伸びる 2 本の前記オフセット推移線分同士の第 1 交点の有無を検出する裏返り検出ステップと、

該第 1 交点で交差する 2 本の前記オフセット推移線分が共に接続する前記差分図形の辺と前記第 1 交点との間の距離に基づいて第 1 交点を、該距離の短い順にソートするソートステップと、ソートされた順に前記第 1 交点を一つ特定し、特定した第 1 交点で交差する 2 本の前記オフセット推移線分が共に接続する前記オフセット図形の裏返り辺の各端部に接続されたオフセット図形の 2 本の隣接辺同士の第 2 交点を算出し、特定した第 1 交点で交差する 2 本のオフセット推移線分が共に接続する前記差分図形の頂点および前記オフセット図形の頂点を前記第 1 頂点列および前記第 2 頂点列から削除し、特定した第 1 交点を第 1 頂点列に差分図形の新たな頂点として記憶すると共に、前記第 2 交点を第 2 頂点列に、前記差分図形の新たな頂点に対応するオフセット図形の新たな頂点として記憶する頂点更新ステップとを含むことを特徴とする。

【 0 0 1 3 】

さらに、請求項 4 記載のコンピュータに実行させるためのプログラムを格納し

たコンピュータが読み取り可能な記憶媒体では、前記頂点更新ステップにおいて、前記隣接辺同士が平行で前記第2交点が存在しない場合には、前記第1交点を通過し、かつ前記隣接辺と平行な第2仮想直線を生成すると共に、2本の隣接辺の非裏返り辺側の各端部に位置するオフセット図形の各頂点から伸びる2本のオフセット推移線分と前記第2仮想直線との第3交点を求め、2本の隣接辺の内、第1交点により近い第3交点を含むオフセット推移線分と接続された隣接辺を前記裏返り辺と合わせ、全体を新たな裏返り辺とすることを特徴とする。

【0014】

これらによれば、裏返り辺の除去処理は、裏返り辺が発生する順、裏返り箇所の近傍のみ裏返り辺の除去処理を行うので、マイナスサイジング処理の処理ステップの内、オフセット処理についてはその計算オーダは、

$$O(N + M \log M) \quad \text{但し、} 0 \leq M \leq N$$

となり、

また、OR処理についてはその計算オーダは、

$$O((N - M) \log(N - M))$$

となる。ここで、Nは元図形の頂点数、Mは裏返り辺数である。

そして、これら計算オーダの合計は、従来例で説明した第1のサイジング処理方法よりも少なく、またオフセットも一度に行うことから第2のサイジング処理のような計算誤差が生ずることもない。

【0015】

【発明の実施の形態】

以下、本発明に係るマイナスサイジング処理方法の好適な実施の形態を添付図面に基づいて詳細に説明する。

一例として図12に示す頂点(p1, p2, p3, p4, p5, p6)で構成される元図形(差分図形)Aに対してマイナスサイジング処理する場合の手順を、図1のフローチャートと、図2～図6を用いて説明する。ここで説明の都合上、元図形Aの各頂点の組(p1, p2, p3, p4, p5, p6)を第1頂点列と呼ぶ。この第1頂点列は予め記憶されている。この元図形Aは前述のように、元図形Aを消滅させるような大きなサイジング量を適用して仮のオフセット図形Bを生成した際に裏返り辺が発生す

る図形である。

【 0 0 1 6 】

まず、元図形 A の各辺上から元図形 A の内方へ仮想直線をサイジング量（オフセット量）分だけ平行移動させ、移動後の仮想直線同士の交点（オフセット点）を求め、記憶する。ここで、オフセット点は、元図形 A の第 1 頂点列（ $p_1, p_2, p_3, p_4, p_5, p_6$ ）に対応して、それぞれ $p_1', p_2', p_3', p_4', p_5', p_6'$ となる。

そして、オフセット後の仮想直線で囲まれて成る図形、具体的には頂点 p_1' と頂点 p_2' を結ぶ線分（ p_1', p_2' ）、 p_2' と p_3' を結ぶ線分（ p_2', p_3' ）、 p_3' と p_4' を結ぶ線分（ p_3', p_4' ）、 p_4' と p_5' を結ぶ線分（ p_4', p_5' ）、 p_5' と p_6' を結ぶ線分（ p_5', p_6' ）および p_6' と p_1' を結ぶ線分（ p_6', p_1' ）が一辺となり、こられ線分で囲まれてなる図形（オフセット図形）B を図 2 に示すように生成する。なお、説明の都合上、オフセット図形 B の各頂点（オフセット点）の組み（ $p_1', p_2', p_3', p_4', p_5', p_6'$ ）を第 2 頂点列と呼ぶ（ステップ S 1 0 0 : オフセット図形生成ステップ）。

【 0 0 1 7 】

次に、元図形 A の各頂点（ $p_1, p_2, p_3, p_4, p_5, p_6$ ）とこの各頂点に対応したオフセット図形 B の各頂点（ $p_1', p_2', p_3', p_4', p_5', p_6'$ ）とを結ぶオフセット推移線分を生成する（オフセット推移線分生成ステップ）。図 3 参照。

オフセット推移線分は、頂点 p_1 と頂点 p_1' を結ぶ線分（ p_1, p_1' ）、 p_2 と p_2' を結ぶ線分（ p_2, p_2' ）、 p_3 と p_3' を結ぶ線分（ p_3, p_3' ）、 p_4 と p_4' を結ぶ線分（ p_4, p_4' ）、 p_5 と p_5' を結ぶ線分（ p_5, p_5' ）および p_6 と p_6' を結ぶ線分（ p_6, p_6' ）である。

また、元図形 A の隣接する各頂点から伸びる 2 本のオフセット推移線分同士の第 1 交点の有無を検出する（裏返し検出ステップ）。図 3 では、オフセット推移線分（ p_2, p_2' ）とオフセット推移線分（ p_3, p_3' ）との間、およびオフセット推移線分（ p_5, p_5' ）とオフセット推移線分（ p_6, p_6' ）との間に第 1 交点 p_{23} 、 p_{56} がそれぞれ検出される。また、併せて裏返し辺（ p_2', p_3' ）と（ p_5', p_6' ）も検出される。

以上、オフセット推移線分生成ステップと裏返し検出ステップとを合わせてス

テップ S 1 0 2 とする。

【 0 0 1 8 】

次に、第 1 交点でそれぞれ交差する 2 本のオフセット推移線分が共に接続する元図形 A の各辺と第 1 交点との間の距離 C に基づいて第 1 交点を、この距離 C の短い順にソートする（ステップ S 1 0 4 : ソートステップ）。第 1 交点をソートするということは、第 1 交点に対応して特定される裏返り辺をもソートすることにもなる。図 3 では、第 1 交点が p23 と p56 の 2 点あり、第 1 交点 p23 と辺 (p2, p3) との間の距離 $C_1 >$ 第 1 交点 p56 と辺 (p5, p6) との間の距離 C_2 であるから、第 1 交点 p56、p23 の順番にソートされる。

【 0 0 1 9 】

次に、ソートされた順番に第 1 交点を一つ特定し、特定した第 1 交点に対応する裏返り辺を除去する処理を、第 1 交点が終了するまで行う。

この処理の内容について詳細に説明する。

まず、最初の第 1 交点 p56 で交差する 2 本のオフセット推移線分 (p5, p5')、(p6, p6') が共に接続するオフセット図形 B の裏返り辺 (p5', p6') の各端部に接続されたオフセット図形 B の 2 本の隣接辺 (p4', p5')、(p6', p1') 同士の交点（第 2 交点）p56' を算出する。

そして、特定した第 1 交点 p56 で交差する 2 本のオフセット推移線分が共に接続する元図形 A の 2 つの頂点 (p5 と p6) およびオフセット図形 B の 2 つの頂点 (p5' と p6') を第 1 頂点列および第 2 頂点列から削除する。そして、特定した第 1 交点 p56 を第 1 頂点列に元図形 A の新たな頂点として記憶すると共に、第 2 交点 p56' を第 2 頂点列に、元図形 A の新たな頂点 p56 に対応するオフセット図形 B の新たな頂点として記憶する。これにより特定した第 1 交点 p56 に対応した裏返り辺 (p5', p6') の除去が行える。（ステップ S 1 0 6 : 頂点更新ステップ）。図 4 参照。よって、第 1 頂点列は (p1, p2, p3, p4, p56) となり、第 2 頂点列は (p1', p2', p3', p4', p56') となる。

【 0 0 2 0 】

次に、更新後の各頂点列から生成されるオフセット推移線分に基づいて新たなオフセット推移線分同士の第 1 交点の有無、つまり裏返り辺の有無を検出する（

ステップ S 1 0 8 : 2 次裏返り辺検出ステップ)。

そして、新たな第 1 交点が検出された場合には、先にソートした第 1 交点と合わせて全体をソートし直し (ステップ S 1 1 0 : 再ソートステップ)、ステップ S 1 0 6 に戻り、最初の第 1 交点に対応した裏返り辺の除去処理を行う。

なお、ステップ S 1 0 8 において、新たな第 1 交点が検出されなかった場合には、ステップ S 1 1 0 をスキップし、ステップ S 1 0 6 に戻る。本実施の形態では、新たなオフセット推移線分は他のオフセット推移線分とは交差せず、新たな第 1 交点は生成されない。

【 0 0 2 1 】

このため、第 1 交点 p_{23} に対して裏返り辺の除去処理を、第 1 交点 p_{56} の場合と同様に続いて行う。

ここで、最初の第 1 交点 p_{23} で交差する 2 本のオフセット推移線分 (p_2, p_2')、(p_3, p_3') が共に接続するオフセット図形 B の裏返り辺 (p_2', p_3') の各端部に接続されたオフセット図形 B の 2 本の隣接辺 (p_1', p_2')、(p_3', p_4') 同士の交点 (第 2 交点) を算出するのであるが、この元図形 A では、辺 (p_1, p_2)、(p_3, p_4) 同士が平行であり、オフセット図形 B の辺 (p_1', p_2')、(p_3', p_4') 同士も平行であり、第 2 交点が存在しない。

【 0 0 2 2 】

このような場合には、オフセット図形 B の裏返り辺 (p_2', p_3') を見直す。

その方法は、一旦裏返り辺更新ステップを実施し、裏返り辺 (p_2', p_3') に隣接する 2 つの隣接辺 (p_1', p_2')、(p_3', p_4') の内の一方を裏返り辺の一部としてから第 2 交点を算出するようにする。

この裏返り辺更新ステップについて説明する。

まず、図 5 のように、第 1 交点 p_{23} を通過し、かつ各隣接辺 (p_1', p_2')、(p_3', p_4') と平行な第 2 仮想直線 L を生成する。

【 0 0 2 3 】

そして、2 本の隣接辺 (p_1', p_2')、(p_3', p_4') の非裏返り辺側の各端部に位置するオフセット図形 B の各頂点 p_1', p_4' から伸びる 2 本のオフセット推移線分 (p_1, p_1')、(p_4, p_4') と第 2 仮想直線 L との第 3 交点 p_{13}, p_{24} を求め

る。

そして、2本の隣接辺 $(p1', p2')$ 、 $(p3', p4')$ の内、第1交点 $p23$ により近い第3交点を含むオフセット推移線分と接続された隣接辺を求め、求めた隣接辺を裏返り辺と合わせて全体を新たな裏返り辺とする。この場合、この条件に合致する第3交点は $p24$ となり、第3交点 $p24$ を含むオフセット推移線分は $(p4, p4')$ であるから、求める隣接辺は $(p3', p4')$ となる。

そして、隣接辺は $(p3', p4')$ を裏返り辺 $(p2', p3')$ と合わせて全体 $(p2', p3', p4')$ を新たな裏返り辺とし、ステップ S 1 0 6 以降の処理ステップに移行する。

【 0 0 2 4 】

ステップ S 1 0 6 以降の処理を簡単に説明する。

まず、オフセット図形 B の裏返り辺 $(p2', p3', p4')$ の各端部に接続されたオフセット図形 B の2本の隣接辺 $(p1', p2')$ 、 $(p4', p56')$ 同士の交点 (第2交点) $p24'$ を算出する。

そして、オフセット図形 B の裏返り辺 $(p2', p3', p4')$ に含まれる頂点 $p2'$ 、 $p3'$ 、 $p4'$ と、裏返り辺 $(p2', p3', p4')$ に接続されるオフセット推移線分 $(p2, p2')$ 、 $(p3, p3')$ 、 $(p4, p4')$ が接続される元図形 A の頂点 $p2, p3, p4$ とを、それぞれ第2頂点列および第1頂点列から削除する。

そして、前述のごとく選択された第3交点 $p24$ を、 $p24'$ に対する元図形 A の新たな頂点として第1頂点列に含めて記憶し、また第2交点 $p24'$ をオフセット図形 B の新たな頂点として第2頂点列に含め記憶する。

【 0 0 2 5 】

以上の処理によって、第1頂点列は $(p1, p24, p56)$ となり、第2頂点列は $(p1', p24', p56')$ となる。図6参照。

そして、この状態では、オフセット推移線分 $(p1, p1')$ 、 $(p24, p24')$ 、 $(p56, p56')$ 同士は互いに交差しないため、裏返り辺が存在しない。

よって、オフセット処理が完了することになる。

なお、ステップ S 1 0 2 において各オフセット推移線分間に第1交点が生じない場合には、裏返り辺が存在せず、裏返り辺の除去処理が不要である。よって、

ステップS104～ステップS110をスキップし、オフセット処理を完了することになる。

【0026】

そして、以上のように裏返り辺の除去処理が施された図6に示すオフセット図形Bに対して、従来例と同様にOR処理を施すことによってマイナスサイジング処理が完了する。

そして、各ステップにおける計算オーダは図1に示すように、

ステップS100・・・ $O(N)$

ステップS102・・・ $O(N)$

ステップS104・・・ $O(M \log M)$

ステップS106・・・ $O(M)$

ステップS108・・・ $O(M)$

ステップS110・・・ $O(M \log M)$

であるから、全体としてのオフセット処理の計算オーダは、

$O(N + M \log M)$

となる。

【0027】

そして、この計算オーダを従来例で説明した図11のマイナスサイジング処理方法の計算オーダと比較すると、

図11の計算オーダは前述のごとく、

オフセット処理： $O(N)$

OR処理： $O((aN) \log(aN))$

このとき $a \geq 2$

であるから、マイナスサイジング処理全体としては、

$O(N + aN \log aN)$ ・・・式(1)

となる。

【0028】

一方、本実施の形態のマイナスサイジング処理方法の計算オーダは下記のようになる。

オフセット処理 : $O(N + M \log M)$

OR処理 : $O((N - M) \log(N - M))$

このとき、 $0 \leq M \leq N$

であるから、マイナスサイジング処理全体としては、

$O(N + M \log M + (N - M) \log(N - M))$

$= O(N + M \log(M/N - M) + N \log(N - M)) \dots \dots \text{式(2)}$

【0029】

次に、式(1)の第2項と式(2)の第2、3項を以下の3つの場合に分けて比較する。

第1に、 $N - M > M$ 、つまり $M < N/2$ の場合 :

式(2)の第2項がマイナスの値になり、式(2) < 式(1) が成り立つ。

第2に、 $N - M = M$ 、つまり $M = N/2$ の場合 :

式(2)の第2項がゼロとなり、式(2) < 式(1) が成り立つ。

【0030】

第3に、 $N - M < M$ 、つまり $M > N/2$ の場合 :

式(2)の第2項は、 $0 \leq M \leq N$ により、 $N \log N$ よりも小さい。そこで、式(2)の第2項を $N \log N$ に置き換えて式(2)の第2項と第3項の和を求めると、

$N \log N + N \log(N - M)$

$= N(\log N(N - M)) \dots \dots \text{式(3)}$

となる。

ここで、場合分けの条件 : $M > N/2$ より、

$N(\log N(N - M)) < 2N(\log(N/\sqrt{2})) \dots \dots \text{式(4)}$

が成り立つ。

そして、式(4)の右辺は、式(1)の第2項よりも値が小さい。よって、式(2) < 式(1) が成り立つ。

【0031】

つまり、全てにおいて式(2) < 式(1) が成り立つので、従来例で説明した図11の場合の計算オーダよりも、本発明の計算オーダの方が小さい。

よって、マイナスサイジング処理に要する計算量が少なく済み、処理速度が高速であると言える。

また、元図形 A の各辺に対するオフセットを一度に行うため、従来例で説明した図 1 3 に示す処理方法のようにオフセット点に含まれる誤差が増大することもない。

【 0 0 3 2 】

上述したサイジング処理方法を実行するプログラムは、図 1 4 に示すように、記憶媒体（フロッピや C D - R O M 等の外部記憶媒体） 5 0 に格納されており、一例として下記の構成のコンピュータ 5 2 にインストールされて C A D 装置（サイジング処理装置）として動作する。

このコンピュータ（サイジング処理装置） 5 2 の概要構成について図 1 5 を用いて説明する。

5 4 は C P U である。C P U 5 4 は、サイジング処理装置 5 2 全体を制御する。

この C P U 5 4 に、バス 5 6 を介して R O M 5 8、R A M 6 0、H D（ハードディスク装置） 6 2、マウスやキーボード等の入力装置 6 4、外部記憶媒体ドライブ装置 6 6 および C R T 等の表示装置 6 8 が接続されている。

【 0 0 3 3 】

C P U 5 4 の制御プログラムは R O M 5 8 に格納されている。

サイジング処理方法を実行するプログラム（サイジング処理プログラムとも言う）は、記憶媒体 5 0 から H D 6 2 にインストール（記憶）される。また、R A M 6 0 には、サイジング処理方法を C P U 5 4 が実行する際の作業領域や、サイジング処理方法を実行するプログラムの一部が記憶される領域が確保されている。

また、H D 6 2 には、入力データや最終データや O S（オペレーティングシステム）等が予め記憶される。

【 0 0 3 4 】

まず、コンピュータ 5 2 の電源を投入すると、C P U 5 4 が R O M 5 8 から制御プログラムを読み出し、さらに H D 6 2 から O S を読み込んで、O S を起動さ

せる。これによりコンピュータ 5 2 はサイジング処理プログラムを記憶媒体 5 0 からインストール可能な状態となる。

次に、記憶媒体 5 0 を外部記憶媒体ドライブ装置 6 6 に装着し、入力装置 6 4 から制御コマンドを CPU 5 4 に入力し、記憶媒体 5 0 に格納されたサイジング処理プログラムを読み取って HD 6 2 等に記憶する。つまりサイジング処理プログラムをコンピュータ 5 2 にインストールする。

その後は、サイジング処理プログラムを起動させると、コンピュータ 5 2 はサイジング処理装置として動作し、オペレータはサイジング処理プログラムが CRT 6 8 に表示する作業内容と手順に従って入力装置 6 4 を操作することで、上述したサイジング処理を行えるようになる。

【 0 0 3 5 】

【発明の効果】

本発明に係るマイナスサイジング処理方法やこの処理をコンピュータに実行させるためのプログラムを格納したコンピュータが読み取り可能な記憶媒体によれば、裏返り辺の除去処理は、裏返り辺が発生する順、裏返り箇所の近傍のみ裏返り辺の除去処理を行うので、マイナスサイジング処理の処理ステップの計算オーダーは比較的少なくて済み、処理時間が短縮できる。また、元図形に対するオフセットも一度に行うことから、元図形の一辺毎にオフセットして裏返り辺の除去処理を行う方法のように計算誤差が生ずることもない、という効果を奏する。

【図面の簡単な説明】

【図 1】

本発明に係るマイナスサイジング処理方法のオフセット処理を説明するためのフローチャートである。

【図 2】

元図形に対するオフセット図形を生成した状態を示す説明図である。

【図 3】

オフセット推移線分同士の第 1 交点、つまり裏返り辺を抽出した状態を示す説明図である。

【図 4】

最初の第 1 交点に対応する裏返り辺の除去を行った状態を示す説明図である。

【図 5】

隣接辺同士が平行な 2 番目の第 1 交点に対応する裏返り辺の除去を説明するための説明図である。

【図 6】

全ての裏返り辺の除去を行った後の元図形とオフセット図形を示す説明図である。

【図 7】

従来の光近接交差補正処理の検証処理の動作フローであり、(a) は光近接交差補正処理全体の動作フローであり、(b) は従来のマイナスサイジング処理方法を利用した異常補正部分検出処理の動作フローである。

【図 8】

従来のマスクデータ変換処理についての検証処理の動作フローである。

【図 9】

サイジング量によっては裏返り辺が発生し、マイナスサイジング処理が正確に行えない元図形の一の例を示す説明図である。

【図 10】

サイジング量によっては裏返り辺が発生し、マイナスサイジング処理が正確に行えない元図形の他の例を示す説明図である。

【図 11】

正確なマイナスサイジング処理が行えるオフセット処理の従来の方法の一例を示す説明図であり、(a) は元図形、(b) はオフセット処理後のオフセット図形を示す。

【図 12】

サイジング量によっては裏返り辺が発生し、マイナスサイジング処理が正確に行えない元図形の他の例を示す説明図である。

【図 13】

正確なマイナスサイジング処理が行えるオフセット処理の従来の方法の他の例を図 12 の元図形を用いて説明した説明図であり、(a) は最初の裏返り発生ま

でのマイナスサイジングを行った状態を示す図であり、(b)は2番目の裏返り発生までのマイナスサイジングを行った状態を示す図であり、(c)は残り量分のサイジングを行った状態を示す図である。

【図 1 4】

サイジング処理装置の外観、およびサイジング処理方法を実施するサイジング処理プログラムが格納された記憶媒体の外観を示す説明図である。

【図 1 5】

図 1 4 のサイジング処理装置の構成を示すブロック図である。

【符号の説明】

S 1 0 0 オフセット図形生成ステップ

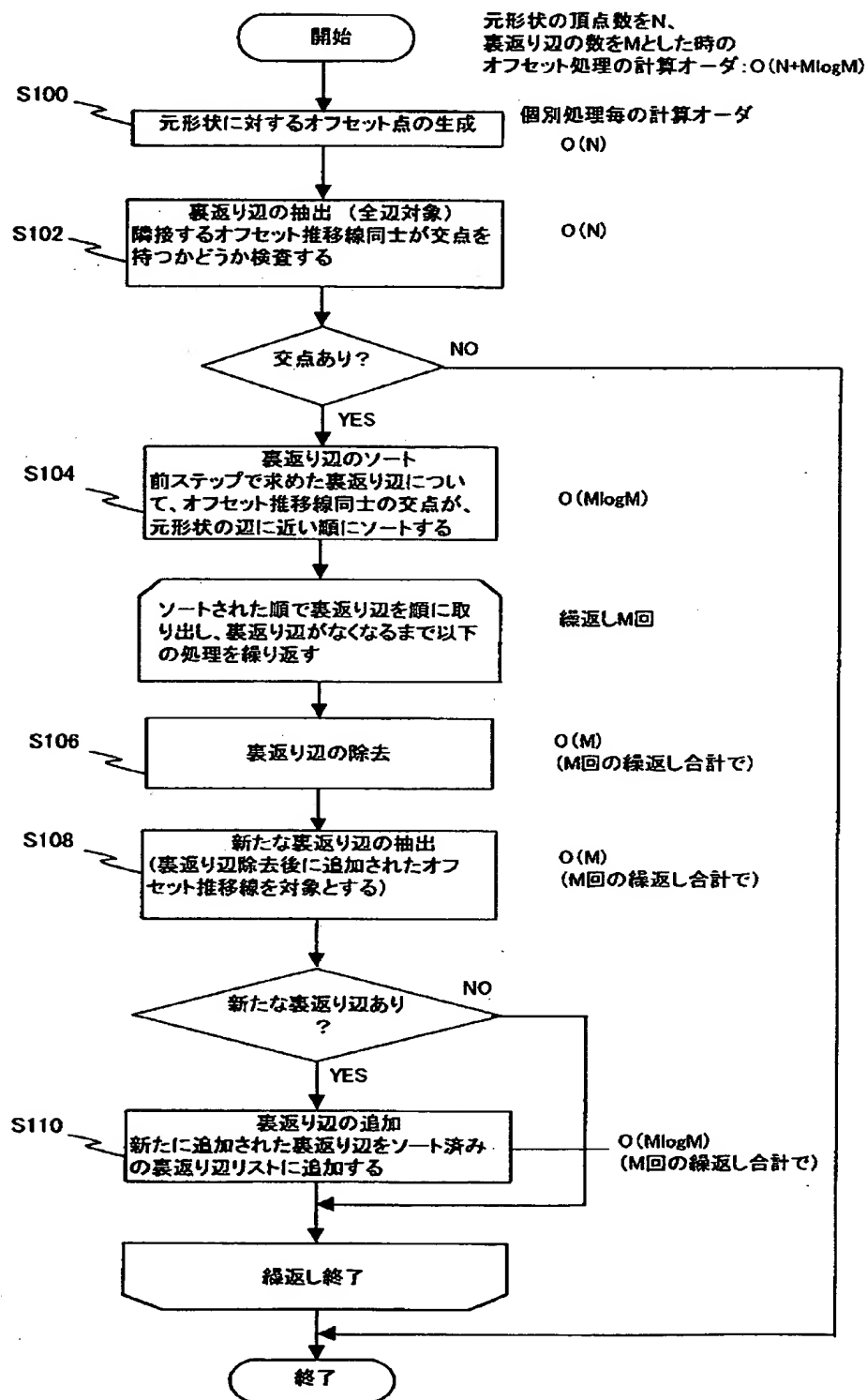
S 1 0 2 裏返り検出ステップ

S 1 0 4 ソートステップ

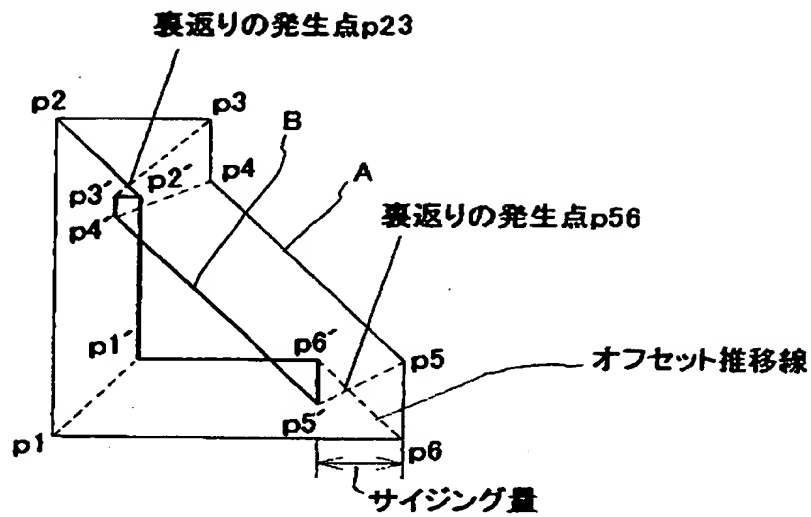
S 1 0 6 ～ S 1 1 0 頂点更新ステップ

【書類名】 図面

【図 1】

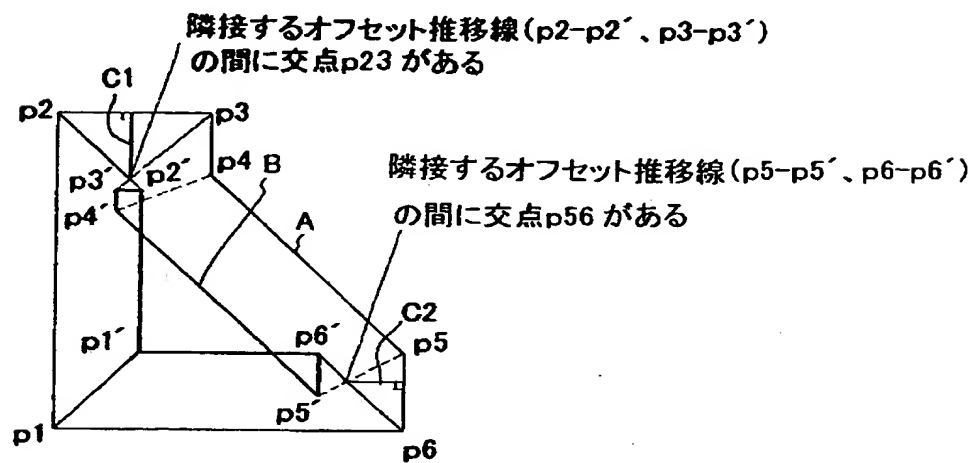


【図 2】



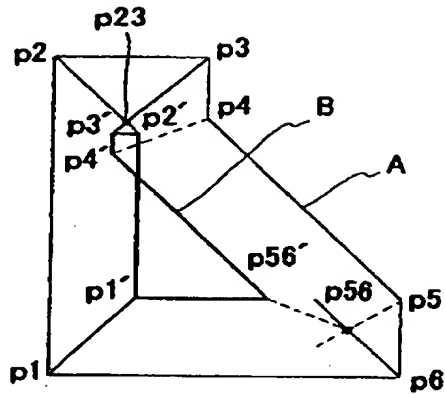
裏返し辺が発生する例

【図 3】



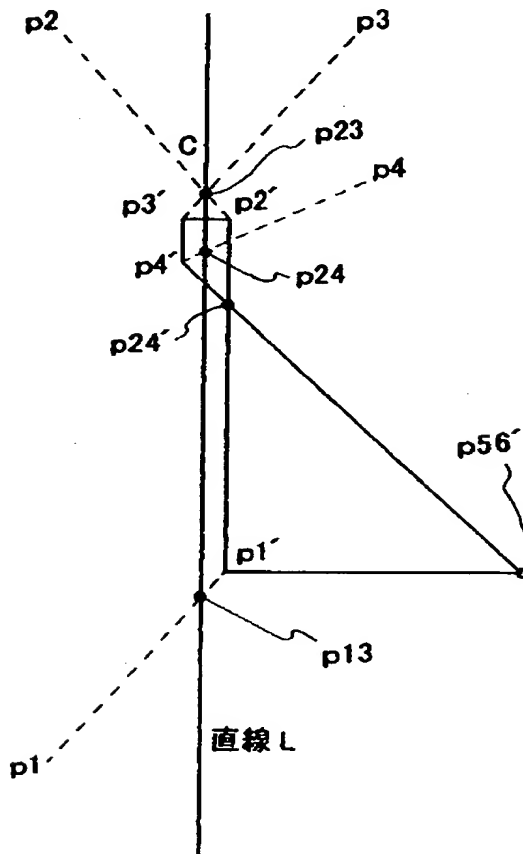
オフセット推移線間の交点の抽出

【図 4】



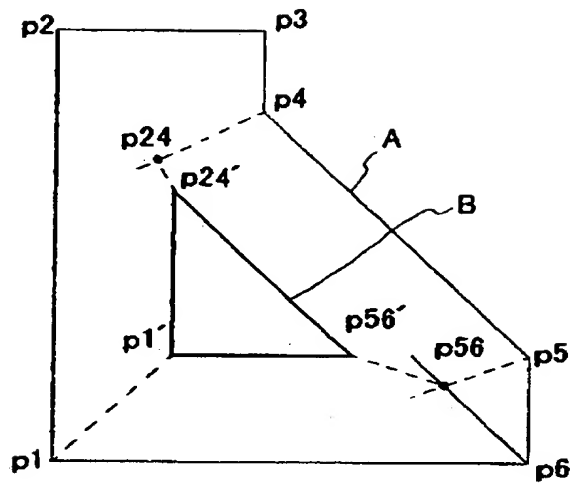
裏返し辺 $p5'-p6'$ の除去

【図 5】



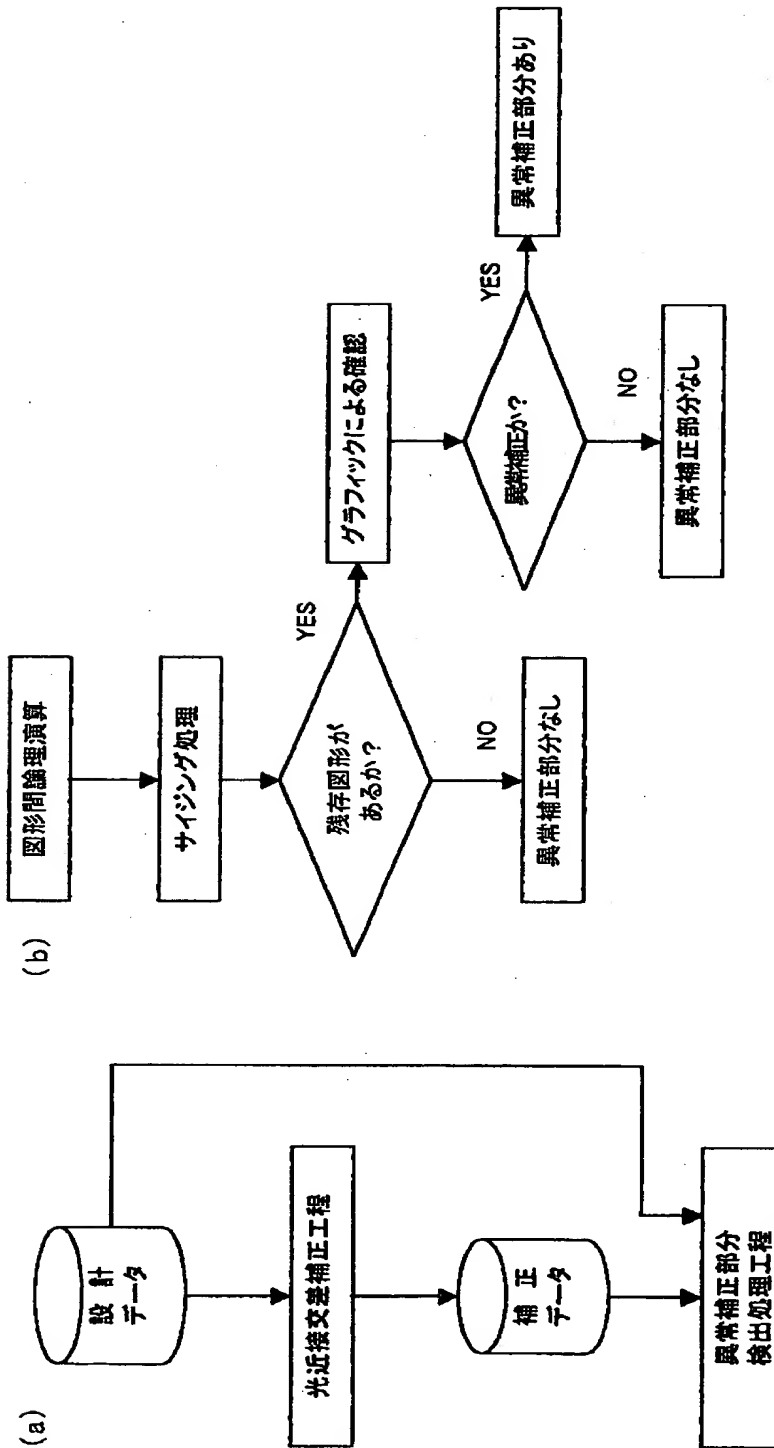
裏返し辺 $p2'-p3'$ の除去(拡大図)

【図 6】

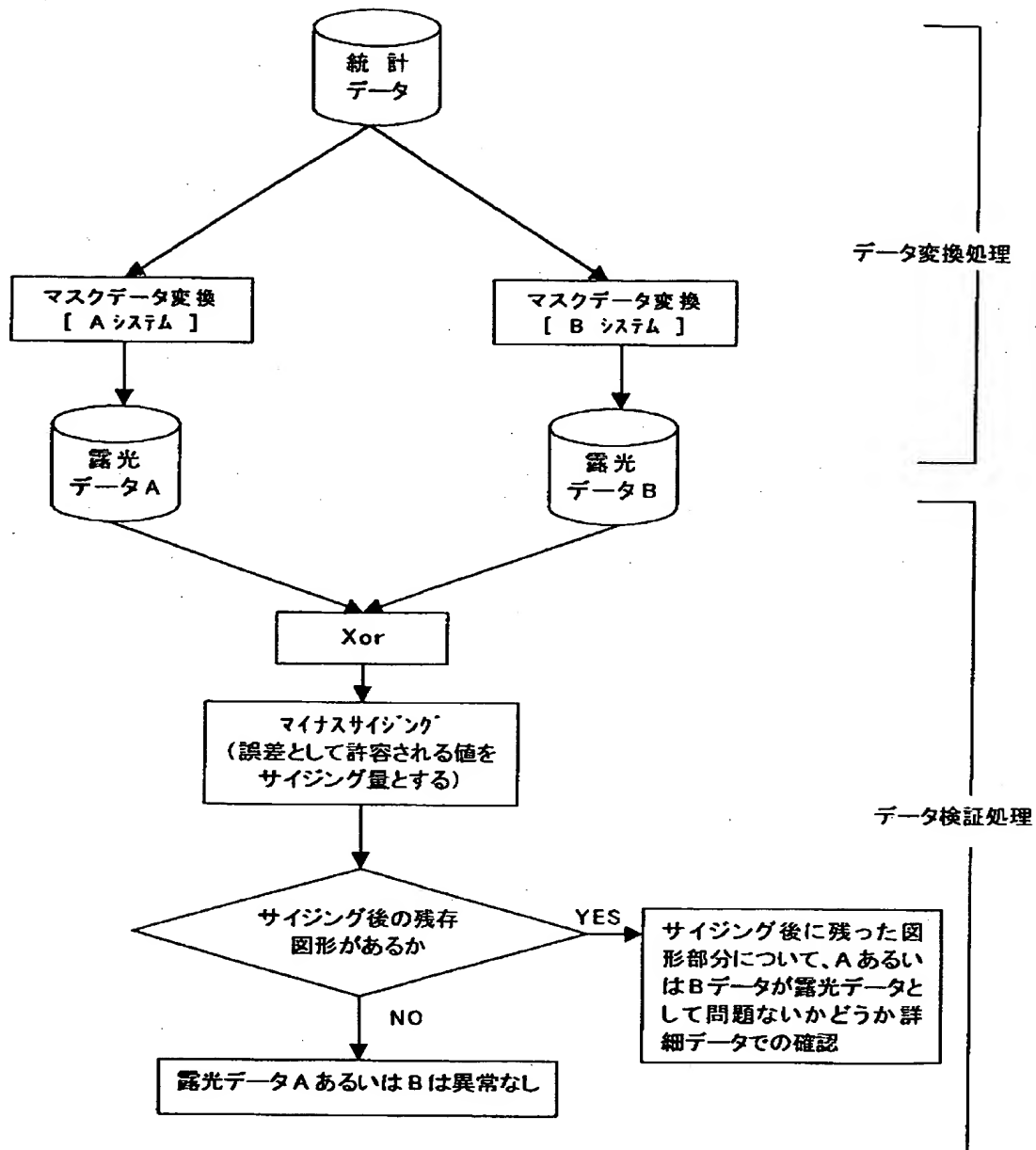


(e) 裏返り辺 $p_2'-p_3'$ の除去後 (最終結果)

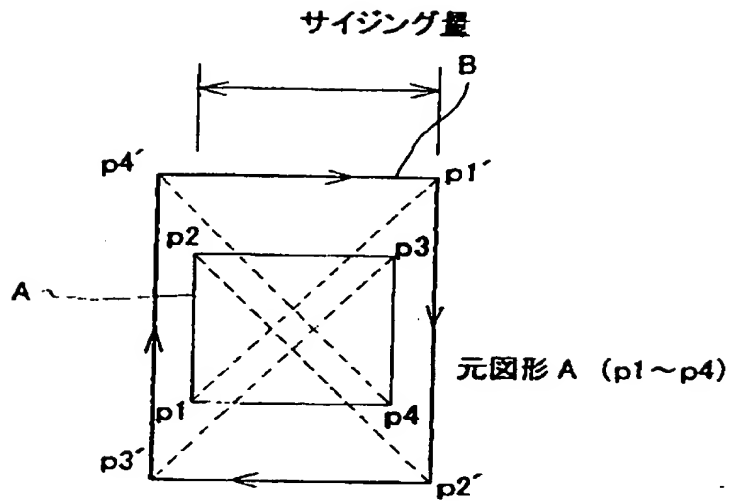
【図 7】



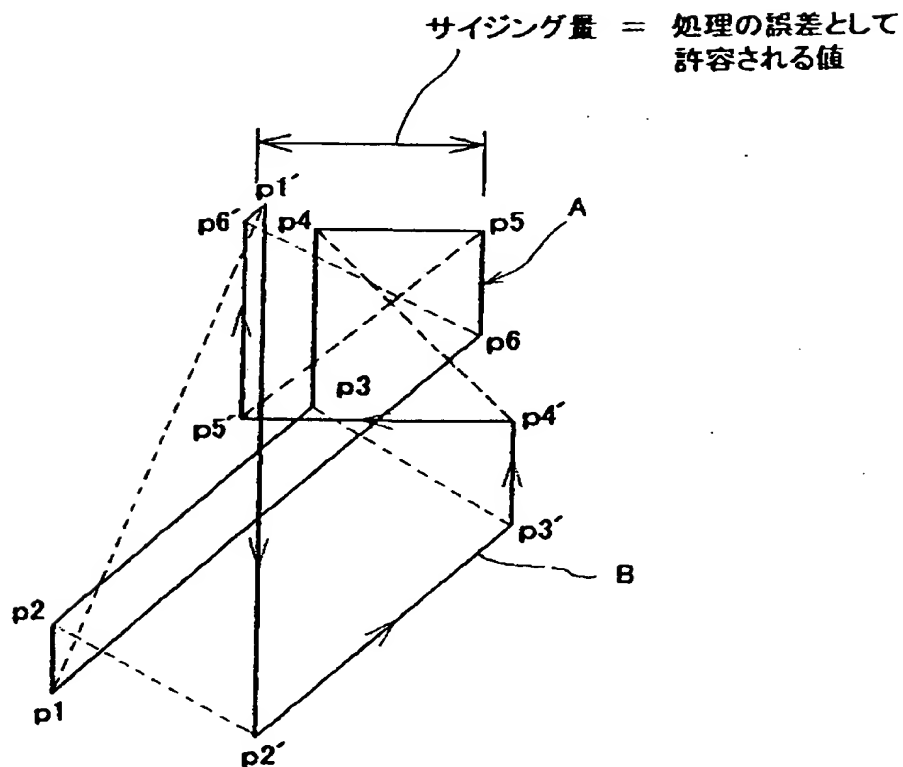
【図 8】



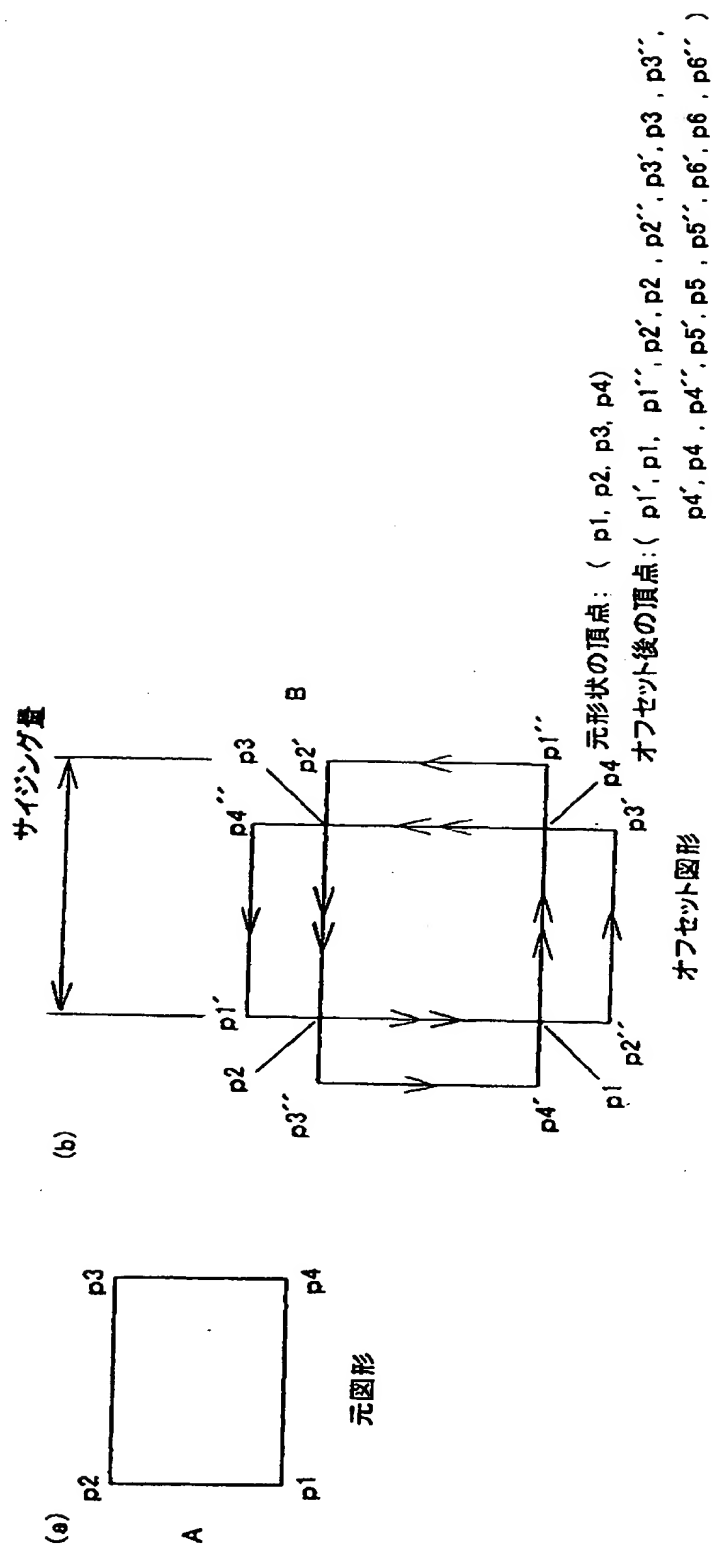
【図 9】



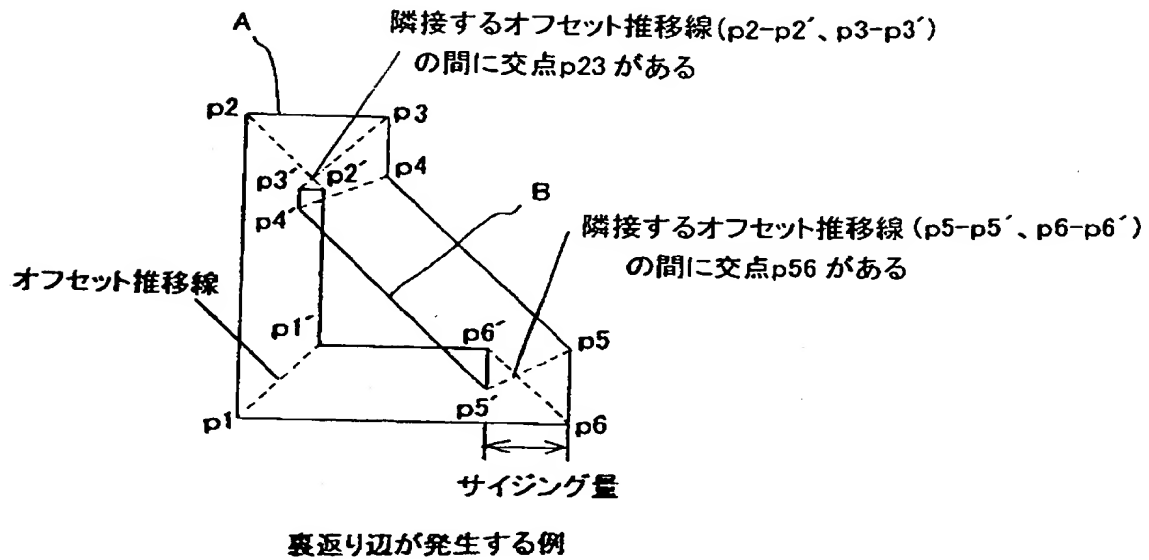
【図 10】



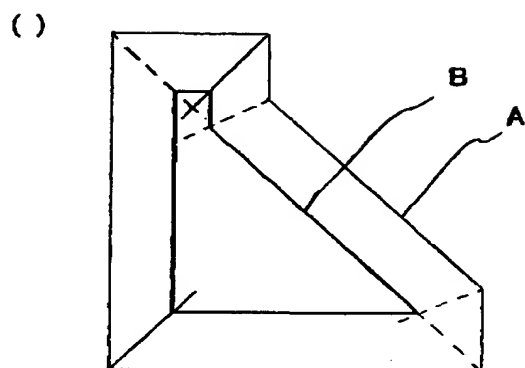
【図 11】



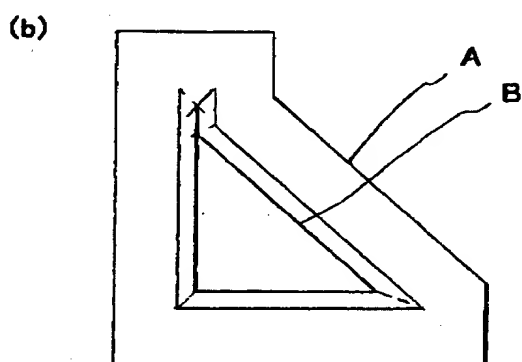
【図 12】



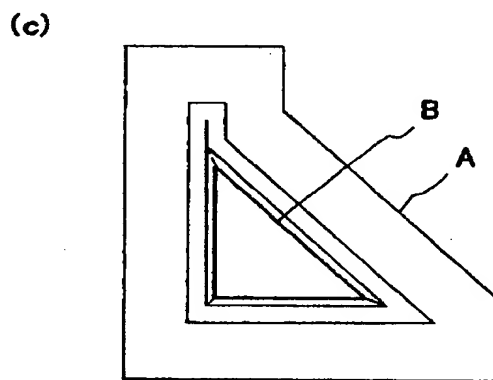
【図 1 3】



最初の裏返し発生 ($p5-p5'$ 、 $p6-p6'$ の交点) までのサイジング

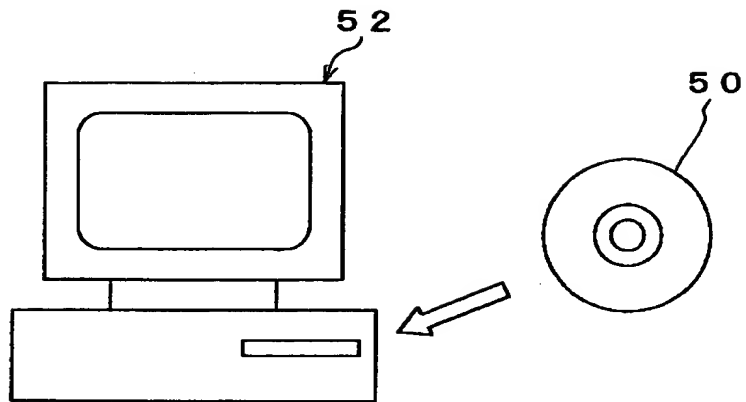


2 番目の裏返し発生 ($p2-p2'$ 、 $p3-p3'$ の交点) までのサイジング

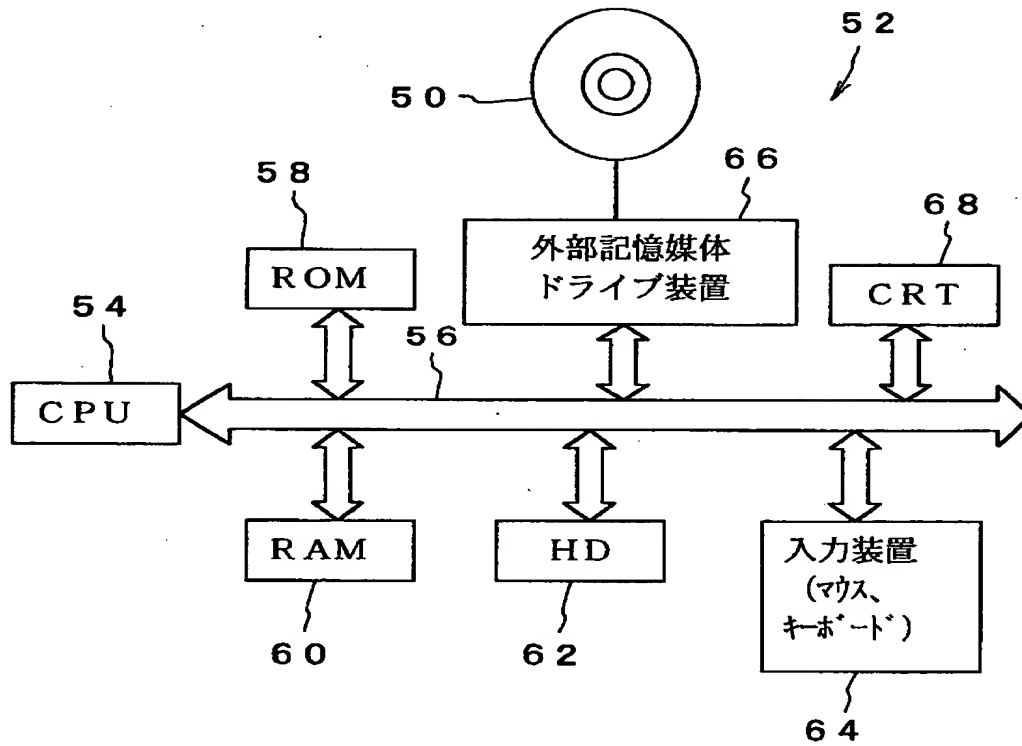


残り量分のサイジング

【図14】



【図15】



【書類名】 要約書

【要約】

【課題】 計算量が少なく高速であり、かつ正確なサイジングが行えるマイナスサイジング処理方法を提供する。

【解決手段】 元図形と図形処理後の新図形間の差分図形の大小により図形処理の検証を行うためのマイナスサイジング処理方法において、差分図形の各頂点を第1頂点列として記憶する。差分図形のオフセット図形を生成して各頂点を第2頂点列として記憶する(S100)。差分図形の各頂点と対応するオフセット図形の各頂点とを結ぶオフセット推移線分を生成し、隣接するオフセット推移線分同士の第1交点の有無を検出する(S102)。第1交点がある場合、交差する2本のオフセット推移線分に対応する差分図形の辺と第1交点との間の距離に基づいて第1交点を距離の短い順にソート(S104)。第1交点のソート順に、この第1交点に対応するオフセット推移線分が共に接続するオフセット図形の裏返り辺の各端部に接続されたオフセット図形の2本の隣接辺同士の第2交点を算出し、特定した第1交点に対応する差分図形およびオフセット図形の頂点を第1頂点列および第2頂点列から削除し、特定した第1交点を第1頂点列に、第2交点を第2頂点列にそれぞれ記憶して裏返り辺を除去する(S106)。

【選択図】 図1

出 願 人 履 歴 情 報

識別番号 [000190688]

1. 変更年月日 1990年 8月20日

[変更理由] 新規登録

住 所 長野県長野市大字栗田字舎利田711番地

氏 名 新光電気工業株式会社